

WEST Search History

DATE: Thursday, October 14, 2004

Hide?	Set Name	Query	Hit Count
		<i>DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L17	L16 and client	95
<input type="checkbox"/>	L16	L8 and 707/100-102.ccls.	152
<input type="checkbox"/>	L15	L14 and L2	35
<input type="checkbox"/>	L14	L12 and L1	261
<input type="checkbox"/>	L13	L12 and (attribute and paramater)	1
<input type="checkbox"/>	L12	L8	3568
		<i>DB=EPAB; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L11	L10	11
		<i>DB=EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L10	L8	445
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L9	L8 and L2	35
<input type="checkbox"/>	L8	L6 and (rule or syntax)	4013
<input type="checkbox"/>	L7	L6 and rule	3418
<input type="checkbox"/>	L6	L5 or l4	20036
<input type="checkbox"/>	L5	program\$ near (creating or creation)	4758
<input type="checkbox"/>	L4	program\$ near (generating or generation)	15815
<input type="checkbox"/>	L3	program\$ near (generat\$ or creat\$)	51830
		<i>DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L2	L1 and (717/107,112,117.ccls.)	246
<input type="checkbox"/>	L1	717/100-123.ccls.	2521

END OF SEARCH HISTORY

Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced
- ☐ CrossRef

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

IEEE Enterprise

- ☐ Access the IEEE Enterprise File Cabinet



Your search matched **12** of **1079782** documents.

A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance** in **Descending** order.

Refine This Search:

You may refine your search by editing the current search expression or entering a new one in the text box.

☐ Check to search within this result set

Results Key:

JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 Automatic test chip and test program generation: an approach to parametric test computer-aided design

Ternisien d'Ouville, T.; Jeanne, J.P.; Leclercq, J.L.; Caloud, D.; Zangara, L.; Microelectronic Test Structures, 1992. ICMTS 1992. Proceedings of the 1992 International Conference on , 16-19 March 1992
Pages:145 - 149

[\[Abstract\]](#) [\[PDF Full-Text \(372 KB\)\]](#) IEEE CNF

2 Expert system for the functional test program generation of digital electronic circuit boards

Lea, S.M.; Brown, N.; Katz, T.; Collins, P.; Test Conference, 1988. Proceedings. 'New Frontiers in Testing', International , 12-14 Sept. 1988
Pages:209 - 220

[\[Abstract\]](#) [\[PDF Full-Text \(948 KB\)\]](#) IEEE CNF

3 The automatic generation of programs for classification problems with grammatical swarm

O'Neill, M.; Brabazon, A.; Adley, C.; Evolutionary Computation, 2004. CEC2004. Congress on , Volume: 1 , 19-23 June 2004
Pages:104 - 110 Vol.1

[\[Abstract\]](#) [\[PDF Full-Text \(530 KB\)\]](#) IEEE CNF

4 XVCL: XML-based variant configuration language

Jarzabek, S.; Bassett, P.; Hongyu Zhang; Weishan Zhang; Software Engineering, 2003. Proceedings. 25th International Conference on , 3-10

May 2003
Pages:810 - 811

[\[Abstract\]](#) [\[PDF Full-Text \(267 KB\)\]](#) IEEE CNF

5 A service logic program generation method for advanced intelligent network

Sakai, H.; Takami, K.; Niitsu, Y.;

Communications, 1996. ICC 96, Conference Record, Converging Technologies for Tomorrow's Applications. 1996 IEEE International Conference on , Volume: 1 , 23-27 June 1996

Pages:47 - 51 vol.1

[\[Abstract\]](#) [\[PDF Full-Text \(500 KB\)\]](#) IEEE CNF

6 IEEE standard VHDL language reference manual

IEEE Std 1076-1987 , 31 March 1988

[\[Abstract\]](#) [\[PDF Full-Text \(8632 KB\)\]](#) IEEE STD

7 Load balancing in a hybrid ATPG environment

Daehn, W.;

Computers, IEEE Transactions on , Volume: 40 , Issue: 7 , July 1991

Pages:878 - 882

[\[Abstract\]](#) [\[PDF Full-Text \(404 KB\)\]](#) IEEE JNL

8 XWRAP: an XML-enabled wrapper construction system for Web information sources

Liu, L.; Pu, C.; Han, W.;

Data Engineering, 2000. Proceedings. 16th International Conference on , 29 Feb.-3 March 2000

Pages:611 - 621

[\[Abstract\]](#) [\[PDF Full-Text \(296 KB\)\]](#) IEEE CNF

9 Set differentiation: a method for the automatic generation of filtering algorithms

Darche, F.D.;

Knowledge-Based Software Engineering Conference, 1996., Proceedings of the 11th , 25-28 Sept. 1996

Pages:134 - 143

[\[Abstract\]](#) [\[PDF Full-Text \(860 KB\)\]](#) IEEE CNF

10 Compiling recursive Prolog programs with list structure into procedural languages

Nam, Y.K.; Henschen, L.J.;

Computer Software and Applications Conference, 1991. COMPSAC '91., Proceedings of the Fifteenth Annual International , 11-13 Sept. 1991

Pages:281 - 287

[\[Abstract\]](#) [\[PDF Full-Text \(568 KB\)\]](#) IEEE CNF

11 **An expert test program generation system for per-pin-testers**

Walter, A.; Kleinman, Y.; Edelshteyn, L.; Gartner, J.;

Test Conference, 1988. Proceedings. 'New Frontiers in Testing'. International , 12-14 Sept. 1988

Pages:665 - 668

[\[Abstract\]](#) [\[PDF Full-Text \(252 KB\)\]](#) IEEE CNF

12 **Simulating von-Neumann machines in an object oriented environment**

Frei, M.;

Applications of Object-Oriented Programming, IEE Colloquium on , 16 Nov 1989

Pages:5/1 - 5/6

[\[Abstract\]](#) [\[PDF Full-Text \(244 KB\)\]](#) IEE CNF

[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) | [New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved

Searching for **generating program and rule**.

Restrict to: [Header](#) [Title](#) Order by: [Expected citations](#) [Hubs](#) [Usage](#) [Date](#) Try: [Google \(CiteSeer\)](#) [Google \(Web\)](#)
[CSB](#) [DBLP](#)

7 documents found. Order: number of citations.

[Hilbert's Building Blocks - Robert Krawczyk Illinois \(1998\) \(Correct\) \(2 citations\)](#)

or interpretation was to be defined in the **program generating** the curve I did not want any manual interpretation or intervention. A set of simple **rules** is used to investigate potential forms. The compositions of NicholasLouis Durand and how simple **rule-based** procedures can be developed to investigate www.iit.edu/~krawczyk/md98.pdf

One or more of the query terms is very common - only partial results have been returned. Try [Google \(CiteSeer\)](#).

[Efficient Simplification of Bisimulation Formulas - Engberg, Larsen \(1995\) \(Correct\) \(1 citation\)](#)

is that the algorithm can be built into the **program generating** the bisimulation formulas and simplify the frequently guarantee that the formula is valid, **rule out** that the formula could be valid, or maybe ftp.imada.ou.dk/pub/papers/pp-1993/47.ps.gz

[Code Generation: Evaluating Polynomials - Fateman \(2000\) \(Correct\)](#)

code. In this paper we describe some **program-generating programs** that may be of assistance. All the code is that it may be worthwhile to use automated **program-generation** programs to write source code. Such be evaluated, is not that much harder, and Horner's **rule** provides a good method. Given a programming www.cs.berkeley.edu/~fateman/papers/polyval.pdf

[Building program generators the easy way \(Extended Abstract\) - Kamin \(Correct\)](#)

Building **program generators** the easy way Extended Abstract) Samuel recursive descent) parsing tells us that this **rule** as a whole produces a function. Thus, the value of produces a function. Thus, the value of such a **rule** has type "Cfunction definition. The **rules** on www-sal.cs.uiuc.edu/~kamin/pubs/ml-meta-extabs.ps

[What Is a Random String? \(Extended Abstract\) - Calude \(Correct\)](#)

Let us call canonical program the smallest **program generating** a string. We claim that every canonical regular sequences, or those in which we observe a **rule** that is easy to grasp, and into irregular www.cs.auckland.ac.nz/~cristian/wir-abstract.ps.gz

[Montages/Gem-Mex: a Meta Visual Programming Generator - Anlauff, Kutter, Pierantonio \(Correct\)](#)

in editing the visual specifications and **generating program** animation and debugging tools. Finally, and universes. The state transitions are given by **rules** that update functions pointwise and extend specifies how they evolve by means of transition **rules**. The initial state is assumed to include the linux.eecs.umich.edu/5/groups/gasm/gemmextr.ps.gz

[Interactive Construction of Decision Models Based on Causal.. - Marek Druzdzal \(Correct\)](#)

are dealt with autonomously by the **generating program**. Breese (1992) proposed a way to represent based on logical knowledge base approach. Facts, **rules**, alternative outcomes, probabilistic base includes a graph grammar as rewriting **rule** and a classification tree to encode abstract www.isp.pitt.edu/~ching/publications/model.ps

Try your query at: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

CiteSeer.IST - Copyright [NEC](#) and [IST](#)


Terms used **generating program rule component**

Found **15,968** of **143,484**

Sort results by


[Save results to a Binder](#)

Try an [Advanced Search](#)

Display results


[Search Tips](#)

Try this search in [The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Generation of visitor components that implement program transformations](#)

R. E. Kurt Stirewalt, Laura K. Dillon

May 2001 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2001 symposium on Software reusability: putting software reuse in context,**
Volume 26 Issue 3

Full text available:  [pdf\(135.87 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The visitor pattern is appealing to developers of program-analysis tools because it separates the design of the data structures that represent a program from the design of software that traverses these structures. Unfortunately, the visitor pattern is difficult to apply when the analysis involves transformation logic that involves multiple program fragments simultaneously. We encountered this problem in our work on the {\proj} project and discovered a novel way to use multiple cooperating v ...

Keywords: amalia, curried visitor framework, lightweight analysis components, program transformations, visitor pattern

2 [A system for program component specification and code generation](#)

Robert P. Brazile

March 1992 **Proceedings of the 1992 ACM/SIGAPP symposium on Applied computing: technological challenges of the 1990's**

Full text available:  [pdf\(589.01 KB\)](#)

Additional Information: [full citation](#), [references](#), [index terms](#)

3 [Retargetable self-test program generation using constraint logic programming](#)

Ulrich Bieker, Peter Marwedel

January 1995 **Proceedings of the 32nd ACM/IEEE conference on Design automation**

Full text available:  [pdf\(60.72 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

4 [Building, modifying and using component generators](#)

Stephen B. Ornburn, Richard J. LeBlanc

May 1993 **Proceedings of the 15th international conference on Software Engineering**

Full text available:  [pdf\(1.32 MB\)](#)

Additional Information: [full citation](#), [references](#)

5 A symmetrical approach to parsing and generation

Marc Dymetman, Pierre Isabelle, Francois Perrault

August 1990 **Proceedings of the 13th conference on Computational linguistics - Volume 3**

Full text available:  [pdf\(680.81 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Lexical Grammars are a class of unification grammars which share a fixed rule component, for which there exists a simple left-recursion elimination transformation. The parsing and generation programs are seen as two dual non-left-recursive versions of the original grammar, and are implemented through a standard top-down Prolog interpreter. Formal criteria for termination are given as conditions on lexical entries: during parsing as well as during generation the processing of a lexical ent ...

6 A transformational approach to generating application-specific environments

David Garlan, Linxi Cai, Robert L. Nord

November 1992 **ACM SIGSOFT Software Engineering Notes , Proceedings of the fifth ACM SIGSOFT symposium on Software development environments, Volume 17 Issue 5**


Full text available:  [pdf\(1.23 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Current software development environments tend to lie at opposite ends of a spectrum: at one extreme are specialized application generators; at the other are general-purpose programming environments. The former provide strong support for system development and reuse, but are costly to build and available only for limited domains. The latter provide weak support, but are generally available and universally applicable. We describe a technique for automating the production of applic ...

7 Invited application paper: re-engineering C++ components via automatic program transformation

Robert L. Akers, Ira D. Baxter, Michael Mehlich

August 2004 **Proceedings of the 2004 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation**

Full text available:  [pdf\(200.37 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Automated program transformation holds promise for a variety of software life cycle endeavors, particularly where the size of legacy systems makes code analysis, re-engineering, and evolution very difficult and expensive. But constructing transformation tools that handle the full generality of modern languages and that scale to very large applications is itself a painstaking and expensive process. This cost can be managed by developing a common transformation system infrastructure that is re-use ...

Keywords: C++, abstract syntax trees, compilers, component architectures, legacy systems, migration, patterns, re-engineering, rewrite rules, software analysis, software transformation

8 On negation in HiLog (extended abstract)

Kenneth A. Ross


April 1991 **Proceedings of the tenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems**

Full text available:  [pdf\(936.56 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

9 Transformation of data traversals and operations in application programs to account for semantic changes of databases

Stanley Y. W. Su, Herman Lam, Der Her Lo

June 1981 **ACM Transactions on Database Systems (TODS)**, Volume 6 Issue 2

Full text available:  pdf(3.00 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper addresses the problem of application program conversion to account for changes in database semantics that result in changes in the schema and database contents. With the observation that the existing data models can be viewed as alternative ways of modeling the same database semantics, a methodology of application program analysis and conversion based on an existing-DBMS-model-and schema-independent representation of both the database and programs is presented. In this methodology ...

Keywords: access pattern, application program conversion, database changes, semantic data model, transformation rules

10 Achieving extensibility through product-lines and domain-specific languages: a case study

April 2002 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 11 Issue 2

Full text available:  pdf(324.37 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This is a case study in the use of *product-line architectures (PLAs)* and *domain-specific languages (DSLs)* to design an extensible command-and-control simulator for Army fire support. The reusable components of our PLA are layers or "aspects" whose addition or removal simultaneously impacts the source code of multiple objects in multiple, distributed programs. The complexity of our component specifications is substantially reduced by using a DSL for defining and refining state machi ...

Keywords: GenVoca, aspects, domain-specific languages, refinements, simulation

11 Type classes in Haskell

Cordelia V. Hall, Kevin Hammond, Simon L. Peyton Jones, Philip L. Wadler

March 1996 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 18 Issue 2

Full text available:  pdf(307.05 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This article defines a set of type inference rules for resolving overloading introduced by type classes, as used in the functional programming language Haskell. Programs including type classes are transformed into ones which may be typed by standard Hindley-Milner inference rules. In contrast to other work on type classes, the rules presented here relate directly to Haskell programs. An innovative aspect of this work is the use of second-order lambda calculus to record type information in t ...

Keywords: Haskell, functional programming, type classes, types

12 A heuristic approach for test case generation

Kai-Hsiung Chang, W. Homer Carlisle, James H. Cross, David B. Brown

April 1999 **Proceedings of the 19th annual conference on Computer Science**

Full text available:  pdf(571.91 KB)

Additional Information: [full citation](#), [references](#)

Keywords: artificial intelligence, rule-based systems, software engineering, software testing

13 Automatic program generation using sequent calculus

Talal Maghrabi, Forouzan Golshani

April 1992 **Proceedings of the 1992 ACM annual conference on Communications**

Full text available:  [pdf\(749.25 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Program development can be made amenable to formal methods by using a logical framework. A logic specification, whose operational semantics is based on proof theory, provides an abstract and "implementation independent" definition of the problem, the data domains and the associated operators. Unlike many of the current efforts in this area that use resolution, our approach is based on natural deduction, more specifically, sequent calculus. Following the methodology pr ...

14 The Vienna Definition Language

Peter Wegner


January 1972 **ACM Computing Surveys (CSUR)**, Volume 4 Issue 1

Full text available:  [pdf\(3.89 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

15 Graph rewrite systems for program optimization

Uwe Assmann

July 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
Volume 22 Issue 4

Full text available:  [pdf\(571.22 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)


Graph rewrite systems can be used to specify and generate program optimizations. For termination of the systems several rule-based criteria are developed, defining exhaustive graph rewrite systems. For nondeterministic systems stratification is introduced which automatically selects single normal forms. To illustrate how far the methodology reaches, parts of the lazy code motion optimization are specified. The resulting graph rewrite system classes can be e ...

Keywords: compiler generators, graph rewrite systems, program analysis, program optimization, program transformation, specification, stratification, very high-level languages, visual programming

16 A practical framework for demand-driven interprocedural data flow analysis

Evelyn Duesterwald, Rajiv Gupta, Mary Lou Soffa

November 1997 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
Volume 19 Issue 6

Full text available:  [pdf\(412.57 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


The high cost and growing importance of interprocedural data flow analysis have led to an increased interest in demand-driven algorithms. In this article, we present a general framework for developing demand-driven interprocedural data flow analyzers and report our experience in evaluating the performance of this approach. A demand for data flow information is modeled as a set of queries. The framework includes a generic demand-driven algorithm that determines the response to query by itera ...

Keywords: copy constant propagation, data flow analysis, def-use chains, demand-driven algorithms, distributive data flow frameworks, interprocedural data flow analysis, program optimizations

17 Program Transformation Systems

H. Partsch, R. Steinbrüggen

September 1983 **ACM Computing Surveys (CSUR)**, Volume 15 Issue 3

Full text available:  [pdf\(3.00 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

18 A mechanism for automatically and dynamically changing software components

Katsuhisa Maruyama, Ken-ichi Shima

May 1997 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 1997 symposium on Software reusability**, Volume 22 Issue 3

Full text available:  [pdf\(1.73 MB\)](#)

Additional Information: [full citation](#), [references](#), [index terms](#)

19 Automating proofs of the absence of common runtime errors

Steven M. German

January 1978 **Proceedings of the 5th ACM SIGACT-SIGPLAN symposium on Principles of programming languages**

Full text available:  [pdf\(1.42 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The Runcheck Verifier is a working system for proving the absence of common runtime errors. The language accepted is Pascal without variant records, side effects in functions, shared variable parameters to procedures, or functional arguments. The errors checked are: 1) accessing a variable that has not been assigned a value, 2) array subscripting out of range, 3) subrange type error, 4) dereferencing a NIL pointer, 5) arithmetic overflow, and 6) division by zero.

20 Tabled evaluation with delaying for general logic programs

Weidong Chen, David S. Warren

January 1996 **Journal of the ACM (JACM)**, Volume 43 Issue 1

Full text available:  [pdf\(4.10 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

SLD resolution with negation as finite failure (SLDNF) reflects the procedural interpretation of predicate calculus as a programming language and forms the computational basis for Prolog systems. Despite its advantages for stack-based memory management, SLDNF is often not appropriate for query evaluation for three reasons: (a) it may not terminate due to infinite positive recursion; (b) it may be terminate due to infinite recursion through negation; and (c) it may repeatedly evaluate the s ...

Keywords: program transformations, stable models, tabled evaluation, well-founded models

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

(19) Organisation Mondiale de la Propriété
Intellectuelle
Bureau international



(43) Date de la publication internationale
17 mai 2001 (17.05.2001)

PCT

(10) Numéro de publication internationale
WO 01/35686 A1

(51) Classification internationale des brevets⁷: H04Q 7/32, G06F 9/445

(72) Inventeur; et

(75) Inventeur/Déposant (pour US seulement): DAMBRE, Antoine (FR/FR); 51, avenue du Général Michel Bizot, F-75012 Paris (FR).

(21) Numéro de la demande internationale:

PCT/FR00/03016

(22) Date de dépôt international:

27 octobre 2000 (27.10.2000).

(74) Mandataire: VIDON, Patrice; Le Nobel, 2, allée Antoine Becquerel, Boîte postale 90333, F-35703 Rennes Cedex 7 (FR).

(25) Langue de dépôt:

français

(26) Langue de publication:

français

(30) Données relatives à la priorité:

99/14322

9 novembre 1999 (09.11.1999) FR

(81) États désignés (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

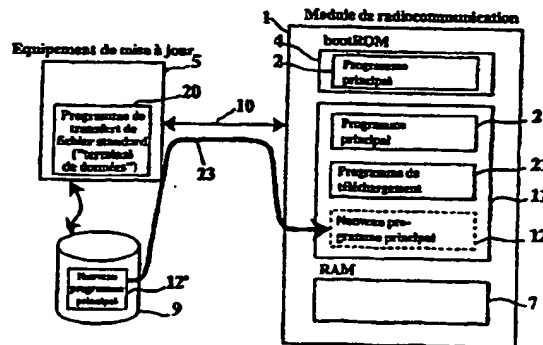
(71) Déposant (pour tous les États désignés sauf US): WAVE-COM [FR/FR]; 39, rue du Gouverneur Général Eboué, F-92130 Issy-les-Moulineaux (FR).

(84) États désignés (régional): brevet ARIPO (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), brevet eurasien

[Suite sur la page suivante]

(54) Titre: METHOD FOR UPDATING A MAIN PROGRAMME EXECUTED BY A RADIO COMMUNICATION MODULE

(54) Titre: PROCEDE DE MISE A JOUR D'UN PROGRAMME PRINCIPAL EXECUTE PAR UN MODULE DE RADIOCOMMUNICATION



- 1...RADIO COMMUNICATION MODULE
- 2...MAIN PROGRAMME
- 21...DOWNLOADING PROGRAMME
- 12...NEW MAIN PROGRAMME
- 5...UPDATING EQUIPMENT
- 20...STANDARD FILE TRANSFER PROGRAMME (DATA TERMINAL)
- 12...NEW MAIN PROGRAMME

(57) Abstract: The invention concerns a method for updating a main programme (2) executed by a radio communication module (1), and/or data associated with said main programme, said radio communication module communicating with an updating equipment (5). The invention is characterised in that it comprises the following successive steps: launching the execution, by the radio communication module, of a downloading programme (21) capable of communicating, in accordance with a standard communication protocol, with a standard file transfer programme (20), of the data terminal type, executed by the updating equipment; sending through the standard file transfer programme (20) to the downloading programme (21), in accordance with the standard communication protocol, a new main programme (12) and/or new data; storing through the downloading programme (21), in a non-volatile memory (11) of the radio communication module (1), the new main module (12) and/or the new data.

[Suite sur la page suivante]

WO 01/35686 A1



Web

Results 21 - 30 of about 64 for "generating a program" rule component method. (0.45 seconds)

[\[PDF\] Porting ROME to a new architecture](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... As well as supplying a **method** for calling ... As a **rule**, the rome module should require no ... rome_start_critical) irrespective of which actual **component** contains the ...

rome.sourceforge.net/downloads/v1.0/docs/PortingGuide.pdf - [Similar pages](#)

[... Continued from Synergistic Market Analysis](#)

... The noise **component** within raw price data tends to ... The **rule**-based portion of the system could ... goes even further by automatically **generating a program** from a ...

www.profitmaker.com/synergy_more.asp - 95k - [Cached](#) - [Similar pages](#)

[\[PDF\] Software Patents at the European Patent Office in 1995](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... pat applicant invention ep0606466 3COM CORP Apparatus for automatic initiation of data transmission ep0748562 AAGAARD MARTENS HARALD METHOD AND APPARATUS FOR ...

swpat.ffii.org/patents/txt/ep/swpikxtep95.en.pdf - [Similar pages](#)

[Psychology of Programming: Annotated Bibliography](#)

... upon whether they are **generating a program** as opposed ... expressiveness purpose of a **component** is readily ... active critic information regarding **rule** violations or ...

www.cise.ufl.edu/research/ParallelPatterns/PatternLanguage/Background/Psychology/Psych-bibliography.htm - 77k - [Cached](#) - [Similar pages](#)

[\[PDF\] FUSE Best Practice Training Materials](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... have defined the interface of each **component**, the part ... take place in several ways:

1. **Generating a program** in a ... guided by the Golden Design Rule (see Appendix A ...

www.fuse-network.com/fuse/training/fpga/fpga.pdf - [Similar pages](#)

[\[PDF\] V5R3 DB2 UDB for iSeries](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... SQL as the common, portable **method** of communicating ... a new source file rather than **generating a program**. ... Procedure Calls - WORF **component** handles invocation ...

www.common.be/pdffiles/02062004UDB.pdf - [Similar pages](#)

[Do You Trust Your Compiler?](#)

... when we say a software **component** is correct ... automate differentiation because determining which **rule** to apply ... such as compiling or **generating a program** from a ...

doi.ieeecomputersociety.org/10.1109/2.762804 - [Similar pages](#)

[\[PDF\] eDeveloper V9 - How To iii Contents](#)

File Format: PDF/Adobe Acrobat

... 215 Adding Settings to a **Component**..... 215 15 Partitioning eDeveloper ...

ftp2.magicsoftware.com/msemirror/www/pdf/devguide.pdf - [Similar pages](#)

[\[PDF\] On Jones-Optimal Specialization for Strongly Typed Languages](#)

File Format: PDF/Adobe Acrobat

... be able to fall back to a cruder **method** without affecting ... is symbolic function

composition - implemented naively by simply **generating a program** whose main ...

www.springerlink.com/index/2M2LRKL4BLHP44WW.pdf - [Similar pages](#)

Microsoft Visual FoxPro: Product Information

... well with other applications through OLE **component** software and ... example, you can create a **rule**, which creates an ... a GENDBC.PRG for **generating a program** that can. ...
www.microsoft.com/vfoxpro/prodinfo/foxmacrg.htm - 77k - [Cached](#) - [Similar pages](#)

◀ Goooooogle ▶

Result Page: [Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [Next](#)

"generating a program" rule compon

[Search within results](#) | [Language Tools](#) | [Search Tips](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)











©2004 Google

Web Images Directory Local **NEW!** News Products

YAHOO! search

Shortcuts Advanced Search Preferences

Search Results Results **1 - 10** of about **141** for **"generating a program" rule component method** - 0.83 sec. (About

1. [EP patents matching keyword 'program'](#) 
... EP679030. Packet program **component** detector. ... Apparatus and **method** for **generating a program** for parall
gauss.bacon.su.se/indices/keyword/ 166 - 525k - [Cached](#) - [More from this site](#)
2. <http://www.sandh.com/snhnew/manuals/BBSNET19.HTM> 
Chapter 19 Web Server/CGI Support and TPL. 19.1 Introduction to CGI ... is compared against the first **rule**. If it ma
this contains ... With the first **method**, the GET **method**, the program examines ...
www.sandh.com/snhnew/manuals/ BBSNET19.HTM - 50k - [Cached](#) - [More from this site](#)
3. <http://www.ibiblio.org/patents/txt/sofpat.txt> 
TITLES TO 2700 SOFTWARE PATENTS ISSUED IN 1992 and 1993 Greg Aharonian Internet Patents News Servic
attenuation data **Rule**-based technique to ... Compound control **method** for controlling a system Electronic **compon**
www.ibiblio.org/patents/txt/ sofpat.txt - 222k - [Cached](#) - [More from this site](#)
4. <http://lpf.ai.mit.edu/Patents/abstracts/2700-list> 
TITLES TO 2700 SOFTWARE PATENTS ISSUED IN 1992 and 1993 Greg Aharonian Internet Patent News Service
attenuation data **Rule**-based technique to ... Compound control **method** for controlling a system Electronic **compon**
lpf.ai.mit.edu/Patents/abstracts/ 2700-list - 227k - [Cached](#) - [More from this site](#)
5. <http://www.sandh.com/snhnew/manuals/cgi01.HTM> 
i Chapter 1 Web Server/CGI Support and TPL. 1.1 Introduction to CGI ... translated is compared against the first rul
rule for this contains ... process the form and the **method**. used to pass ...
www.sandh.com/snhnew/manuals/ cgi01.HTM - 23k - [Cached](#) - [More from this site](#)
6. [EP patents published 1990](#) 
... EP355377. **Method** for testing optically flat electronic **component** assemblies. ... EP355716. **Rule** generating and
program voltage. ...
gauss.bacon.su.se/indices/year/1990 - 346k - [Cached](#) - [More from this site](#)
7. [Senior Community Service Employment Program \[Proposed Rules\] \[04/28/2003\]](#) 
... Service Employment Program; Proposed **Rule** [[Page 22520 ... any proposed project **component** if it believes tha
selecting ...
www.dol.gov/eta/regs/fedreg/ proposed/2003009579.htm - 312k - [Cached](#) - [More from this site](#)
8. <http://lpf.ai.mit.edu/Patents/abstracts/395700.1990-1992> 
Patent Number 4901231 Issue Date 1990 02 13 Appl. Data 944586 1986 12 22 Assignee American Telephone and
Bishop, Thomas P. ... preprocessing Abstract A **method** of assuring consistency of ... computer system and **metho**
component parts, and then encoding the ...
lpf.ai.mit.edu/Patents/abstracts/ 395700.1990-1992 - 274k - [Cached](#) - [More from this site](#)
9. [Software Patents at the European Patent Office in 1998](#) 
... 3COM CORP. **Method** and apparatus for transparent intermediate system ... PATENT GMBH. **Method** of messag
method for a dynamic ...
swpat.ffii.org/patents/txt/ep/ 1998.en.html - 524k - [Cached](#) - [More from this site](#)
10. [Patents Ex Parte Decision \(O/401/02\) \(PDF\)](#) 
... GB 0026184.2 entitled, "**Component** Software", was filed on ... c) a scheme, **rule** or **method** for performing a me

Results Page:

1 [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [Next](#)

[Web](#) | [Images](#) | [Directory](#) | [Local](#) ^{NEW!} | [News](#) | [Products](#)

Your Search:

Help us improve your search experience. [Send us feedback.](#)

Create your own personal search experience with [My Yahoo! Search](#) [BETA]

Copyright © 2004 Yahoo! Inc. All rights reserved. [Privacy Policy](#) - [Terms of Service](#) - [Submit Your Site](#)